

In the Claims:

Please amend claims 1, 9, 14 and 24 as indicated below.

1. (Currently amended) A system, comprising:

a server cluster, comprising:

a plurality of server nodes, wherein each server node comprises:

a server container;

one or more applications configured to execute within the server container, wherein the one or more applications are configured to process requests from a plurality of users as sessions, wherein each session comprises a series of requests from a same user, wherein the one or more applications access and modify application state data during the processing of a given one of the sessions, wherein the application state data comprises session state data for the given one of the sessions; and

a Java Data Object (JDO) persistence manager configured to detect changes to the application state data within the server container and to persist the application state data; and

a persistent data store coupled to the cluster, configured to store application state data of the one or more applications of each respective server container, and configured to make the application state data accessible to each of the plurality of server nodes;

wherein one of the applications of one of the plurality of server nodes is configured to access from the persistent data store application state data for a session that has previously been processed on another one of the plurality of server nodes;

wherein in response to detecting a change in application state data within the server container, the JDO persistence manager is configured to persist only a changed portion of the application state data within the respective server container to the persistent data store.

2. (Previously presented) The system as recited in claim 1, wherein the JDO persistence manager is configured to persist only mutated application state data to the data store, only in response to mutation of the application state data.

3. (Original) The system, as recited in claim 1, wherein the application state data comprises hypertext transfer protocol (http) session data.

4. (Original) The system, as recited in claim 1, wherein the application state data comprises a session bean.

5. (Original) The system as recited in claim 1, further comprising a JDO-style write barrier configured to detect mutation of the application state data.

6. (Original) The system as recited in claim 1, wherein one or more of the applications is configured to function as a distributed application across two or more of the server nodes.

7. (Original) The system as recited in claim 1, wherein the plurality of server nodes is configured to detect the failure of a cluster node and recover sessions from a failed node by accessing session state data from the persistent data store.

8. (Original) The system as recited in claim 1, further comprising a non-sticky load balancer configured to distribute session requests to server nodes based on server workload, wherein the persistence mechanism is configured to synchronize session data to the persistent store.

9. (Currently amended) A system, comprising:

an application server, comprising;

an application server container;

one or more applications configured to execute within the application server container, wherein the one or more applications are configured to process requests from a plurality of users as sessions, wherein each session comprises a series of requests from a same user, wherein the one or more applications access and modify application state data during the processing of a given one of the sessions, wherein the application state data comprises session state data for the given one of the sessions; and

a Java Data Object (JDO) persistence manager configured to detect changes to the application state data within the application server container and to persist the application state data; and

a persistent data store coupled to the application server, configured to store application state data of the one or more applications, and configured to make the application state data accessible to the application server and one or more other application servers;

wherein one of the one or more applications is configured to access from the persistent data store application state data for a session that has previously been processed on one of the one or more other application servers;

wherein in response to detecting a change in application state data within the application server container, the JDO persistence manager is configured to persist only a changed portion of the application state data within the application server container to the persistent data store.

10. (Previously presented) The system as recited in claim 9, wherein the JDO persistence manager is configured to persist only mutated application state data to the data store, only in response to mutation of the application state data.

11. (Original) The system, as recited in claim 9, wherein the application state data comprises hypertext transfer protocol (http) session data.

12. (Original) The system, as recited in claim 9, wherein the application state data comprises a session bean.

13. (Original) The system as recited in claim 9, further comprising a JDO-style write barrier configured to detect mutation of the application state data.

14. (Currently amended) A method, comprising:

a Java Data Object (JDO) persistence manager detecting an access to application state data within a server, wherein application state data is accessed by an application executing on the server during the processing of a given one of a plurality of sessions, wherein the application is configured to process requests from a plurality of users as sessions, wherein each session comprises a series of requests from a same user, wherein the application state data comprises session state data for the given one of the sessions;

the JDO persistence manager determining whether the access alters the application state, in response to said detecting; and

in response to determining that the access alters the application state within the server, the JDO persistence manager persisting only the elements of the application state that are changed by the access to a persistent store that makes the application state accessible to the server and to one or more other servers;

wherein the application is configured to access from the persistent data store application state data for a session that has previously been processed on one of the one or more other servers.

15. (Previously presented) The method, as recited in claim 14, wherein the JDO persistence manager persisting comprises persisting only mutated application state data to the data store, only in response to mutation of the application state data.

16. (Original) The method, as recited in claim 14, wherein the application state data comprises hypertext transfer protocol (http) session data.

17. (Original) The method, as recited in claim 14, wherein the application state data comprises a session bean.

18. (Original) The method, as recited in claim 14, wherein said determining is performed by a JDO-style write barrier configured to detect mutation of the application state data.

19. (Original) The method, as recited in claim 14, wherein said application state data comprises state data for one or more applications configured to function as a distributed application across two or more server nodes of a cluster.

20. – 23. (Canceled)

24. (Currently amended) A computer-accessible storage medium storing program instructions, wherein the program instructions are computer-executable to implement a Java Data Object (JDO) persistence manager configured to:

detect an access to application state data within a server, wherein application state data is accessed by an application executing on the server during the processing of a given one of a plurality of sessions, wherein the application is configured to process requests from a plurality of users as sessions, wherein each session comprises a series of requests from a same user, wherein the application state data comprises session state data for the given one of the sessions;

in response to said detecting, determine whether the access alters the application state; and

in response to determining that the access alters the application state within the server, persist only the elements of the application state that are changed by the access to a persistent store that makes the application state accessible to the server and to one or more other servers;

wherein the application is configured to access from the persistent data store application state data for a session that has previously been processed on one of the one or more other servers.

25. (Previously presented) The computer-accessible storage medium as recited in claim 24, wherein the JDO persistence manager is configured to persist only mutated application state data to the data store, only in response to mutation of the application state data.

26. (Previously presented) The computer-accessible storage medium as recited in claim 24, wherein the application state data comprises hypertext transfer protocol (http) session data.

27. (Previously presented) The computer-accessible storage medium as recited in claim 24, wherein the application state data comprises a session bean.

28. (Previously presented) The computer-accessible storage medium as recited in claim 24, wherein a JDO-style write barrier detects mutation of the application state data.

29. (Previously presented) The computer-accessible storage medium as recited in claim 24, wherein said application state data comprises state data for one or more applications configured to function as a distributed application across two or more server nodes of a cluster.

30. – 33. (Canceled)